A Semi-Supervised Algorithm for Indonesian Named Entity Recognition

Rezka Aufar Leonandya
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
rezka.aufar@ui.ac.id

Bayu Distiawan
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
b.distiawan@cs.ui.ac.id

Nursidik Heru Praptono
Faculty of Computer Science
Universitas Indonesia
Depok, Indonesia
heru.pra@cs.ui.ac.id

Abstract—**Named Entity Recognition or NER is one of the sub-research field of Information Extraction which can be used for machine translation, question answering, semantic web, etc. One of the biggest challenge of NER is the adversity to construct a manually labeled training data. In this work, we present a semi-supervised approach for Indonesian language NER which is capable of creating high quality training data automatically. Semi-supervised approach works by utilizing unlabeled data made from Wikipedia and DBPedia to form high accuracy and non-redundant additional training data for each iteration of semi-supervised process. We show that our system manages to generate new training data and gain an increasing F1 score as the iteration of semi-supervised process goes.**

Keywords-component; named entity recognition; semi-supervised; stanford ner; wikipedia; dbpedia

I. Introduction

Named Entity Recognition (NER) is the task of finding names such as organizations, persons, locations, etc in text. NER is often identified as a sequence classification problem since whether or not a word or a phrase is classified as having named entity are determined by mostly the context of the sentence [13]. NER is also one of the sub-research field of Information Extraction. It has broad applications in machine translation [2], question answering [3], semantic web [1], and populating knowledge, such as entities [10]. Given an input sentence, an NER classifier identifies words or phrase that are part of a named entity, and assigns the entity type.

Indonesian Named Entity Recognition development has already been done before [4, 5]. Based on prior research, a large amount of human-annotated data are required to develop a high accurate NER classifier, which is expensive to obtain. Sometimes it also led the NER classifier not to be optimal in result.

Liao and Veeramachaneni [12] introduced a semi-supervised algorithm for NER in 2009. Their semi-supervised algorithm shows great result compared to supervised algorithm in several cases. More importantly, it does not need a large amount of initial training data to earn great result as it generates more training data from unlabeled data as the iteration goes. One approach to achieve these goals is to select unlabeled data that has been classified with low confidence score by the classifier trained on the original training data.

Therefore, in this research, we are trying to implement a method to create high quality training data automatically using semi-supervised algorithm [9] by utilizing Wikipedia and DBPedia resource that are already available online. We develop a semi-supervised method which is heavily influenced from Liao and Veeramachaneni's algorithm [12] with an adaptation in the independent decision rule on Indonesian language. Our semi-supervised method consists of iterative process which generates additional training data to be added to our previous training data in order to train new classifier on each iteration. We expect that every classifier trained on each iteration will have higher F1-score than the previous trained classifier. Furthermore, our semi-supervised method is also preceded by an initial supervised process, which serves as the initial F1-score to be compared with the classifier trained on the semi-supervised iteration.

Contributions made in this work include:

- *Tidier training data* than previous experiment [4] with some unnecessary character removed, such an infobox, etc.
- *Twenty five thousand lines of manually tagged testing data*. The purpose of creating a manually tagged testing data is to determine our system performance with human performance.
- *Experimental evaluation* on Semi-Supervised Algorithm for Indonesian Named Entity Recognition. The purpose of the experiment is to generate new training data automatically every iteration, to observe if the training data affects overall F1-score for the same testing data, and also to observe whether our semi-supervised algorithm is in fact increases the F1-score of our NER classifier.

This paper is organized as follows. Section 2 describes the resource that we use in this experiment i.e. Wikipedia Dump and DBPedia, and how do we construct dataset from our resources. Section 3 details on the theoretical foundation of this work i.e. Conditional Random Field, Semi-Supervised Algorithm, and Evaluation method. Section 4 introduces our Semi-Supervised Indonesian NER system including each step on how the system works. The details of experiment and evaluation are explained in section 5. Finally, we conclude our work in section 6.

978-1-4673-8501-5/15 $31.00 © 2015 IEEE
DOI 10.1109/ISCBI.2015.15

II. Resources

Using references from previous research, we use two main resources for this research, Indonesian version of Wikipedia Dump and DBPedia's instance type file [4]. Wikipedia Dump will provide the raw text for training and testing data while the DBPedia instance type will serve as the entity information of words or phrase from the raw text.

A. Wikipedia Dump

Wikipedia Dump is a crowdsourced Wikipedia article that can be downloaded freely on its website[1]. Thus, we assume it is a rich source for named entity because it is created from heterogenous contributors, various writing style, and also contains diverse personal views in describing something in an article. On May 2015, there are approximately 358,000 wikipedia articles in Indonesian language.

However, wikipedia is based on human work, so there is still some errors that need to be cleaned in order to get a proper dataset. For example, there are still some articles in Indonesian (Bahasa) written in English just by translation. These kinds of articles are then excluded from our dataset.

After we get a proper dataset, we need to find a word or phrase which is categorized as named entity in Indonesian language. Typically, we do the named entity labelling manually. But in this research, we will look up the information of which words or phrase categorized as Person, Place, or Organization on DBPedia.

B. DBPedia

DBPedia is a community that provides structrured informations of Wikipedia articles by extracting it from Wikipedia [7]. DBPedia Indonesia is a web application which provide various structured informations from Wikipedia Indonesia [11]. Indonesian version of DBpedia describes 140,993 entities. Those entities consist of 19,567 Persons, 57,702 Places, 5,773 Organisations, and 10,711 Works [11]. For comparison, the English version of DBPedia has 4,004,478 entities.

Unlike prior research which utilized the DBPedia Webservice to retrieve the named entity label for a words/phrase in the raw text, we use the ontology information in the instance type file that can be downloaded from its website. The ontology in the instance file refers to particular title in Wikipedia article and it contains the ontology type of the particular title. This information will help us to give a named entity on candidate phrase which produced by Wikipedia Dump.

However, this information is availbale only when there is already a structured information in DBPedia. Let us say there is an article named Soekarno under the URL http://id.wikipedia.org/wiki/Soekarno in Wikipedia page and there exists one entity named Soekarno on DBPedia page under the URL http://id.dbpedia.org/resource/Soekarno, then the ontology information are already available in the instance type file.

With the instance type file, we can label every words or phrase in wikipedia dump with its named entity. If a word or phrase is categorized as article title, then we can use the ontology information to extract the ontology type as our named entity to label the words or phrase in wikipedia dump. This process creates our training and testing set. Throughout this paper, we call this process "automatic tagging".

However, we also encountered a major problems. The instance type file that we use in this research is not as rich as we thought. There are many words and phrase that should be categorized as named entity, but the automatic tagging failed to do so because the instance type file do not contains the structured information for those words. For example, Jokowi should be categorized as Person, yet our system categorized it as "O" because the instance type file do not have any information on Jokowi. We presume that this thing happened because the contributors for DBPedia Indonesia simply do not update the file to the latest article title, or the word itself simply do not exist as article title thus there are no information about it. Throughout the paper, we will call this problem as "automatic tagging problems".

III. Preliminaries

This section describes the underlying theoretical foundation of semi-supervised Indonesian NER. It consists of three main parts: Conditional Random Field, Semi-Supervised Algorithm, and Evaluation Method.

A. Conditional Random Field

Conditional Random Field is an undirected graphical model trained to maximize the conditional probability of a sequence of labels given the corresponding input sequence [6, 12]. It was first introduced on 2001 by Andrew McCallum and Fernando Pereira [6]. In this research, the sequence of labels refers to named entity and the input sequence refers to a sentence. Let $X, X = x_1 \ldots x_n$, be an input sequence and $Y, Y = y_1 \ldots . y_n$, be the label sequence for the input sequence. The conditional probability of Y given X, $P(Y|X)$ is:

$$P(Y|X) = \frac{1}{Z(X)} \exp(\sum_{n=1}^{N} \sum_{k} \pi_k f_k(y_{n-1}, y_n, X, n)) \qquad (1)$$

$Z(X)$ is a normalization which made $P(Y|X)$ valued between 0 and 1. $Z(X)$ is the sum of all possible label sequence for an input sequence, which is:

$$\sum_{\vec{y} \epsilon Y} \exp(\sum_{n=1}^{N} \sum_{k} \pi_k f_k(y_{n-1}, y_n, X, n)) \qquad (2)$$

f_k is a list of feature function, which often takes a binary value. For example, feature function in linear-chain CRF typically is a function that takes 4 input, such as:

- y_n as label or named entity at n.
- y_{n-1} as label or named entity at *n-1*.
- X as input sequence.
- n as the position of the word.

π_k is a learned weight associated with feature function f_k. The parameters for learned weight can be learned by

[1] http://dumps.wikimedia.org/

calculating gradient ascent of CRF's error function, which is $\log P(Y|X)$.

$$\pi_k : \frac{\partial}{\partial w_k} \log P(Y|X) \quad (3)$$

B. *Semi-Supervised Algorithm*

One of the benefits of semi-supervised learning in Named Entity Recognition is that it involves the utilization of unlabeled data to mitigate the effect of insufficient labeled data on classifier accuracy, which will likely to happen if we do manual tagging on raw text that will take a lot of time.

```
Given:
        L – a small set of labeled training data
        U - unlabeled data
Loop for k iterations:
        Train a classifier $C_k$ based on L
        Extract new data D based on $C_k$
        Add D to L
```

Figure 1. Semi-Supervised Learning Algorithm

Semi-supervised learning essentially attempts to automatically generate high quality training data from an unlabeled corpus.

Based on Liao & Veeramachaneni's research [12], we implement the same idea of their semi-supervised learning technique with some adaptation to Indonesian language. The semi-supervised algorithm will be used to generate additional training data and add it to original training data to create a classifier with some new knowledge based on the additional training data every iteration. The iteration is repeated until the result of the new classifer is stagnant or less than the prior classifier. Figure 1 shows the process of semi-supervised algorithm.

Furthermore, we also implement a method based on Liao & Veeramachaneni's research called independent decision rule to the extracted data D. Independent decision rule is a method for selecting unlabeled data that has been classified with low confidence or with label "O" by the classifier trained on the original training data, but their labels are known with high precision from a pre-defined decision rule.

In this research, we define independent decision rule based on the fact that entities such as organizations, places, etc., have some word that is highly indicative of its class. For example, "Gunung Merapi" phrase have an indicative word "Gunung" that strongly indicates it being categorized as place. If "Gunung Merapi" is known with high precision by the classifier trained on the original data, then we can use the indicative word "Gunung" to search for low-confidence phrase or "O" labelled phrase that starts with "Gunung" in unlabeled data to change its label to "Place". This rule is an adaptation of Liao and Veeramachaneni's rule for English language.

C. *Evalution Method*

In this research, we use F1 score to measure the performance of every classifier trained with semi-supervised learning. If current trained classifier generates less accuracy than its previous, then we stop its training iteration. Otherwise, we continue its training iteration. Equation (4) shows how to calculate F1 score, using harmonic mean of precision and recall.

$$F1 = 2\,\frac{Precision * Recall}{Precision + Recall} \quad (4)$$

IV. Semi-Supervised Indonesian Ner

This section will describe the system of semi-supervised Indonesian NER. The system includes the process of cleaning the data from Wikipedia dump, creating the automatically tagged data for Indonesian NER training documents, training process with Stanford CRF-NER, generating additional training data with semi-supervised algorithm, and testing the classifier. Figure 2 shows the entire system of semi-supervised Indonesian NER.

A. *Pre-processing*

The Wikipedia Dump are available on XML Format which still contains unnecessary content such as Infobox, tables, images, etc.. For this experiment, we only need the paragraph in each articles, so we have to remove unnecessary content from wikipedia dump to retrieve the raw texts. Our re-processing phase consists of three steps, xml extraction, format adjusting, and language filtering.

XML extraction is the step to retrieve the clean paragraph text from Wikipedia Dump. We use WikiExtractor[2] tools developed by Giuseppe Attardi and Antonio Fuschetto from University of Pisa to extract the paragraph from Wikipedia Dump.

After we have a clean paragraph data, the next step is to adjust its format to match stanford-ner requirements. The first adjustment is to make sure that there should only be one sentence for each line and every sentence should be ended by dot character(.). We use NLTK[3] tools to make sure that there is only one sentence for each line. This step is needed for the language filtering step.

We also need to filter our articles. If there exist one articles which does not have at least one named entity according to our dbpedia instance file, then it should be removed.

The last step of pre-processing is language filtering. Language has major role in content removing consideration. Although we are using an Indonesian version of Wikipedia Dump, there is still a possibility that some articles were written in another language than Bahasa Indonesia. For some cases, the article is simply a copied English version of Wikipedia or partially written in another language with some other part is written in Indonesian. To make sure that the Indonesian NER classifier will only consume the Indonesian articles, we will remove all sentence which is not written in Bahasa Indonesia. To identify the language of a sentence, we use a language detector [8].

B. *Entity Tagger and Entity Selector*

At this point, the produced text needed to be tokenized and to be tagged by its named entity. do not provide the

[2] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

[3] http://nltk.org

phrases which likely to contain named entity. For tokenizing, we use the library from stanford-ner and for entity tagging, we build our own script. For every words/phrase we can find in the article, we match it with our dbpedia instance file. If there exists an ontology information for that particular words/phrase, then we will label it with its corresponding label. We will label words with "O" label if there is no ontology information exists for those particular words.

After that, we use this tagged and tokenized file to select our training data. We select 4 training scenarios of 50, 500, 2000, and 5000 minimal named entity for each Person, Place, and Organization in order to enrich our experiment. We begin by picking random sentences and count the named entity that exists in the sentence until each Person, Place, and Organization label reach the minimum quantity.

C. *Initial Training/Supervised Learning*

Having been able to automatically tag the Wikipedia articles, the next process is to make initial Indonesian NER classifier. The initial classifier serves as a first measure for the semi-supervised process. Being trained on original training data only, it gives us the result of supervised learning, which can be compared to the result of the semi-supervised process.

The classifier was developed by Stanford NER[4] tools. This tools use a general implementation of Conditional Random Field (CRF) sequence model. In building the Indonesian NER Model, we only use the default configuration shared by Stanford University. We have not experimented on adjusting any parameter or features on this tool.

D. *Semi-Supervised Learning Algorithm*

Semi supervised learning algorithm is used to generate additional training data from unlabeled data to be added to the original training data in order to increase the accuracy of our classifier. As explained in Theoretical Foundation in Figure 2, we implement an adaptation version of of Liao and Veeramachaneni's semi-supervised algorithm.

The first step of semi-supervised algorithm is to find the probability of each word or phrase from unlabeled data that has been tagged by the classifier. For this step, we uses the stanford-ner library to generate the highest probability label for every word. Stanford-ner use a general implementation of Viterbi to label each word. After that, we determine a threshold to classify whether a word or phrase has a high confidence or not. We set the threshold to 0.98, 0.90, and 0.85 to enrich our experiment variable.

After having every word labeled with its named entity and its probability, the next process is to use independent decision rule on words with low confidence score by utilizing labels from words with high confidence. The following list describes the independent decision rule for each labels:

- Organization

If a sequence of words has been classified as "Organization" with high confidence score, we force the labels of other occurences of the same sequence in the unlabeled data, to be "Organization" and add the sentence containing the sequence of words to the additional training data. Also, if the sequence of words classified with high confidence score contains an indicative words of organization, we use these indicative words to search for other sequence of words starting with the indicative words and change the label to "Organization" if the other sequence of words are classified with low confidences or classified as O.

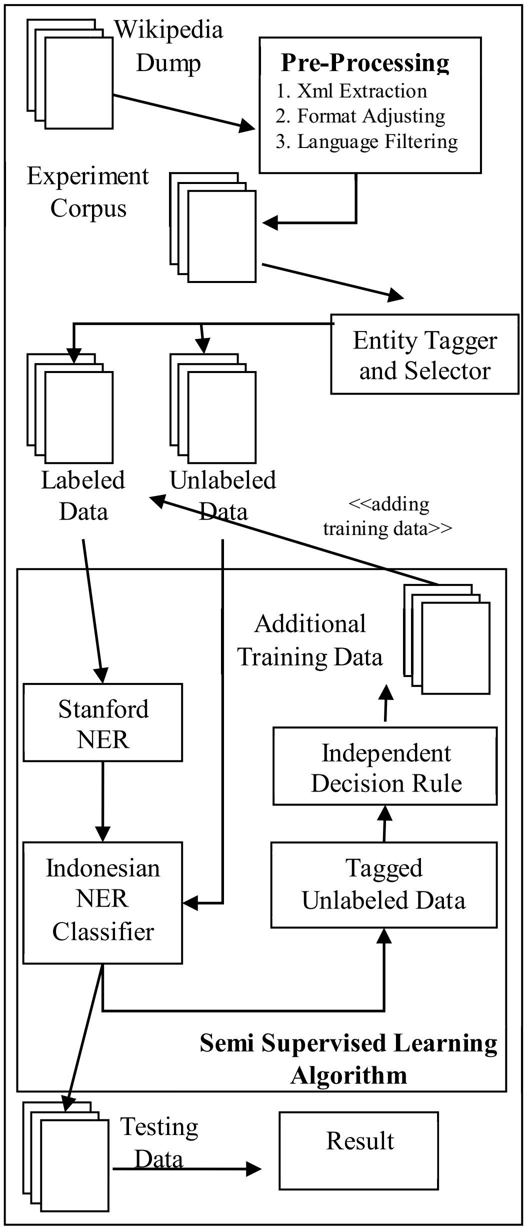

Figure 2. Semi-Supervised Indonesian NER System Overview

[4] http://nlp.stanford.edu/software/stanford-ner-2014-01-04.zip

Example 1:

High-confidence ORG: ***Universitas Indonesia*** *menempati peringkat satu Universitas terbaik di Indonesia.*

ORG with O label: ***Universitas Brawijaya*** *terletak di kota Malang.*

Indicative word: ***Universitas***

Changed label : ***Universitas Brawijaya***

- Place

If a sequence of words has been classified as "Place" with high confidence score, we force the labels of other occurences of the same sequence in the unlabeled data, to be "Place" and add the sentence containing the sequence of words to the additional training data. Also, if the sequence of words classified with high confidence score contains an indicative words of organization, we use these indicative words to search for other sequence of words starting with the indicative words and change the label to "Place" if the other sequence of words are classified with low confidences or classified as O.

Example 2:

High-confidence PLAC: Ridwan Kamil merupakan walikota ***Kota Bandung****.*

PLAC with O label: ***Kota Jakarta*** *disebut-sebut sebagai kota termacet di dunia.*

Indicative word: ***Kota***

Changed label : ***Kota Jakarta***

- Person

If a sequence of words has been classified as "Person" with high confidence score, we force the labels of other occurences of the same sequence in the unlabeled data, to be "Person" and add the sentence containing the sequence of words to the additional training data. Also, if the sequence of words classified as "Person" consists of two or more words, then we will split these words into one word to find the same occurences and change its label into Person.

Example 3:

High-confidence PERS: ***Joko Widodo*** *merupakan presiden ke-7 Republik Indonesia.*

PERS with O label: ***Joko*** *akhirnya memutuskan untuk menjadi pengemudi PT GO-JEK.*

Changed label: ***Joko***

- O

There is not any special rule for O label. However, if we find a high confidence score for every word in a sentence, eventhough all the label is O, we will add it to additional training data. This is needed to maintain the classifier's knowledge based on its previous learning.

V. EXPERIMENTS AND EVALUATION

We divide the experiment into two parts. The first part is the experiment with supervised learning, which the classifier learns from original training data only. The second part is the experiment with semi-supervised learning. Having a classifier from supervised learning, we use it to tag unlabeled data and extract high-quality additional training data to be added to training data for the next training iteration.

To measure the accuracy of the system, we perform a test to a gold standard testing data to see how well the results of the semi-supervised learning compared to supervised learning are. We create a gold standard by taking 75 random Wikipedia documents and manually tag ecah entity in the article.

For the training documents, we collected 12.000 articles (12 MB) in Indonesian that have been automatically tagged with three entities, namely Person, Place, and Organization. From these data, we conducted several experiments to determine the effect of each experiment variables that we determine. For the supervised learning, we build four different NER classifier with 50, 500, 2000, and 5000 minimum named entity training data in order to determine the effect of the amount of training data used. Figure 3 illustrates the result of our supervised learning. There is an increase in F1 score as we add more named entity information to training data.

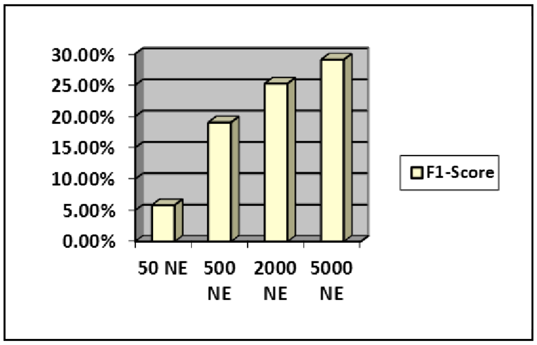

Figure 3. Supervised Learning or Initial Training Result

As for the semi-supervised learning, we add two more experiments variable, which is the value of threshold used (0.98, 0.90, 0.85), and the uses of independent decision rule. Table 1 illustrates the results of semi-supervised learning implemented after supervised learning. As shown in the table, we obtain an increase in F1 score especially in the experiment that uses independent decision rule. The highest increse obtained is 5,43% on 50 named entity training data with 0.85 threshold and using independent decision rule. Overall, we get a low result on F1 score but as the iteration goes, we get an increasing F1 score with semi-supervised algorithm. However, there is also some anomaly in the experiment with 2000 named entity, where the implementation of independent decision rule results in decreasing F1 score at the 4th iteration. This is because for the 2000 and 5000 named entity training data, we only do randomized pick for the training data once, unlike the 50 and 500 named entity training data, where we do randomized pick for three times then use the mean of the F1 score to measure the results. Hence, the results of the classifier trained with 50 and 500 named entity are more representative than the classifier trained with 2000 and 5000 named entity.

VI. CONCLUSION AND FUTURE WORK

Semi-supervised learning shows a better result than the supervised learning as the iteration goes. This is due to an additional knowledge that the classifier gets from decision rule, unlike the supervised process which is trained without additional knowledge. Overall, our system managed to achieve an increasing F1 score as the iteration of the semi-supervised algorithm progresses. However, the F1 score results of the classifier is still low.

For future work, we will find a way to overcome the "automatic tagging problem" in order to increase our initial F1 score. We will also find a way to enrich our decision rule by adding some Bahasa Indonesia rule to indicate entities more exquisitely.

TABLE I. SUPERVISED LEARNING AND INITIAL TRAINING RESULT

Supervised Learning and Initial Training Result					
Experiment	***Iter 0***	***Iter 1***	***Iter 2***	***Iter 3***	***Iter 4***
50 NE CRF HI WR	5,83%	6,94%	8,31%	8,84%	8,88%
50 NE CRF HI NR	5,83%	6,75%	7,40%	6,92%	
50 NE CRF ME WR	5,83%	8,09%	10,04%	10,39%	10,73%
50 NE CRF ME NR	5,83%	6,41%	7,32%	7,40%	7,36%
50 NE CRF LO WR	5,83%	8,17%	10,40%	11,07%	11,26%
50 NE CRF LO NR	5,83%	6,25%	7,14%	7,31%	7,56%
500 NE CRF HI WR	19,03%	20,20%	21,66%	22,12%	22,76%
500 NE CRF HI NR	19,03%	18,65%			
500 NE CRF ME WR	19,03%	20,60%	22,24%	22,79%	22,94%
500 NE CRF ME NR	19,03%	18,99%			
500 NE CRF LO WR	19,03%	20,49%	22,40%	22,60%	23,02%
500 NE CRF LO NR	19,03%	18,79%			
2000 NE CRF HI WR	25,24%	26,17%	27,48%	27,17%	
2000 NE CRF HI NR	25,24%	25,62%	27,46%	27,99%	27,89%
2000 NE CRF ME WR	25,24%	26,20%	27,66%	28,41%	29,03%
2000 NE CRF ME NR	25,24%	26,42%	27,66%	28,08%	28,03%
2000 NE CRF LO WR	25,24%	26,49%	27,99%	28,47%	29,13%
2000 NE CRF LO NR	25,24%	26,32%	27,99%	28,10%	27,87%
5000 NE CRF HI WR	29,08%	29,29%	30,01%	30,70%	30,97%
5000 NE CRF HI NR	29,08%	29,40%	29,49%	29,80%	29,19%
5000 NE CRF ME WR	29,08%	30,44%	31,53%	31,79%	31,83%
5000 NE CRF ME NR	29,08%	30,06%	30,88%	30,97%	31,08%
5000 NE CRF LO WR	29,08%	30,78%	31,54%	31,88%	31,96%
5000 NE CRF LO NR	29,08%	30,11%	30,93%	31,27%	31,14%

REFERENCES

[1] A. Caputo, P. Basile, and G. Semera. (2009). Boosting a Semantic Search Engine by Named Entites. Proceedings of the 18th International Symposium on Foundations of Intelligent Systems (ISMIS), (pp. 241-50). Prague.

[2] B. Babych and A. Hartley. (2003). Improving Machine Translation Quality with Automatic. *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT*, (pp. 1- 8).

[3] C. Lee, Y.-G. Hwang, and M.-G. Jang. (2007). Fine-Grained Named Entity Recognition and Relation Extraction for Question Answering. SIGHR'07 Proceedings (pp. 799-800). Amsterdam: ACM.

[4] F. Rashel, A. Luthfi., B. Distiawan, and R. Manurung. "Building Indonesian Named Entity Recognition using Wikipedia and DBPedia". 2014.

[5] I. Budi, S. Bressan, G. Wahyudi, Z.A. Hasibuan, B.A. A. Nazief. "Named Entity Recognition for the Indonesian Language: Combining Contextual, Morphological and Part-of-Speech Features into a Knowledge Engineering Approach". Discovery Science 2005: 57-69.

[6] J. Lafferty, A. McCallum, and F.Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". 2001.

[7] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P.v. Kleef, S. Auer, C. Bizer. "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia". 2014

[8] N. Shuyo. "Language Detection Library for Java". 2010. http://code.google.com/p/language-detection/

[9] O. Chapelle, B. Schlkopf, and A. Zien. (2010). Semi-Supervised Learning. The MIT Press.

[10] P. Cimiano, A. Madche, S. Staab, and J. Volker. (2009). Ontology Learning. Berlin: Springer

[11] R.A. Prasetya. "Pengembangan DBPEDIA Indonesia". 2013. 56-58

[12] W. Liao, & S. Veeramachaneni. "A Simple Semi-supervised Algorithm For Named Entity Recognition". Proceedings of the NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing. 2009. 58-65.

[13] Z. Zhang. "Named Entity Recognition - Challenges in Document Annotation, Gazetteer Construction and Disambiguation. Sheffield: The University of Sheffield". 2013